Your search matched **11** of **393** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enterin new one in the text box.

((((automatically<in>ab) <and> (generate<in>ab) <

Search

☐ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

---

1 **Estimation and enhancement of real-time software reliability throu mutation analysis**
Geist, R.; Offutt, A.J.; Harris, F.C., Jr.;
Computers, IEEE Transactions on , Volume: 41 , Issue: 5 , May 1992
Pages:550 - 558

[Abstract] [PDF Full-Text (764 KB)] **IEEE JNL**

---

2 **Automatic generation of self-scheduling programs**
Foster, I.;
Parallel and Distributed Systems, IEEE Transactions on , Volume: 2 , Issue: 1 1991
Pages:68 - 78

[Abstract] [PDF Full-Text (976 KB)] **IEEE JNL**

---

3 **Automated detection of injected faults in a differential equation sol**
Last, M.; Friedman, M.;
High Assurance Systems Engineering, 2004. Proceedings. Eighth IEEE Interna Symposium on , 25-26 March 2004
Pages:263 - 264

[Abstract] [PDF Full-Text (206 KB)] **IEEE CNF**

---

4 **Compiler scheduling of mobile agents for minimizing overheads**
Xiaotong Zhuang; Pande, S.;
Distributed Computing Systems, 2003. Proceedings. 23rd International Confer on , 19-22 May 2003
Pages:600 - 607

[Abstract] [PDF Full-Text (312 KB)] IEEE CNF

5 **Generating structured program instances with a high degree of loca**
Leopold, C.;
Parallel and Distributed Processing, 2000. Proceedings. 8th Euromicro Worksh on , 19-21 Jan. 2000
Pages:267 - 274

[Abstract] [PDF Full-Text (92 KB)] IEEE CNF

6 **Using SDL for implementing a wireless medium access control prot**
Hannikainen, M.; Knuutila, J.; Hamalainen, T.; Saarinen, J.;
Multimedia Software Engineering, 2000. Proceedings. International Symposium on , 11-13 Dec. 2000
Pages:229 - 236

[Abstract] [PDF Full-Text (692 KB)] IEEE CNF

7 **A declarative approach for designing and developing adaptive components**
Boinot, P.; Marlet, R.; Noye, J.; Muller, G.; Consel, C.;
Automated Software Engineering, 2000. Proceedings ASE 2000. The Fifteenth International Conference on , 11-15 Sept. 2000
Pages:111 - 119

[Abstract] [PDF Full-Text (764 KB)] IEEE CNF

8 **Compiling object-oriented programs for distributed execution**
Muthukrishnan, P.K.; Bryant, B.R.; Zwarico, A.E.;
High Performance Computing, 1996. Proceedings. 3rd International Conferenc on , 19-22 Dec. 1996
Pages:49 - 54

[Abstract] [PDF Full-Text (464 KB)] IEEE CNF

9 **Evolution support by homogeneously documenting patterns, aspect and traces**
Sametinger, J.; Riebisch, M.;
Software Maintenance and Reengineering, 2002. Proceedings. Sixth European Conference on , 11-13 March 2002
Pages:134 - 140

[Abstract] [PDF Full-Text (517 KB)] IEEE CNF

10 **Verification and validation of high integrity software generated by automatic code generators**
Malepati, V.; Li, H.; Pattipati, K.R.; Deb, S.; Patterson-Hine, A.;
Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on , Volume: 3 , 11-14 Oct. 1998
Pages:3004 - 3009 vol.3

[Abstract] [PDF Full-Text (584 KB)] IEEE CNF

11 **Near fine grain parallel processing of circuit simulation using direc method**

Maekawa, Y.; Nakano, K.; Takai, M.; Kasahara, H.;

Communications, Computers, and Signal Processing, 1995. Proceedings. IEEE Pacific Rim Conference on , 17-19 May 1995

Pages:272 - 276

[Abstract] [PDF Full-Text (396 KB)] **IEEE CNF**

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

Your search matched **0** of **1123491** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enterin new one in the text box.

(dependency<in>ab) <and> (generation<in>ab) <and

Search

☐ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

---

Results:
No documents matched your query.

Your search matched **95** of **1123491** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enterin new one in the text box.

(dependency<in>ab) <and> (generation<in>ab) <and Search

☐ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

---

1 **Modeling correlation in software recovery blocks**
Tomek, L.A.; Muppala, J.K.; Trivedi, K.S.;
Software Engineering, IEEE Transactions on , Volume: 19 , Issue: 11 , Nov. 19
Pages:1071 - 1086

[Abstract] [PDF Full-Text (1304 KB)] **IEEE JNL**

---

2 **A process for software requirements safety analysis**
Mojdehbakhsh, R.; Subramanian, S.; Vishnuvajjala, R.; Tsai, W.-T.; Elliott, L.;
Software Reliability Engineering, 1994. Proceedings., 5th International Sympo on , 6-9 Nov. 1994
Pages:45 - 54

[Abstract] [PDF Full-Text (556 KB)] **IEEE CNF**

---

3 **ERP requirements engineering practice: lessons learned**
Daneva, M.;
Software, IEEE , Volume: 21 , Issue: 2 , March-April 2004
Pages:26 - 33

[Abstract] [PDF Full-Text (254 KB)] **IEEE JNL**

---

4 **Synthesis of multivariate Gaussian random processes with a preassigned covariance (Corresp.)**
Eby, E.;
Information Theory, IEEE Transactions on , Volume: 16 , Issue: 6 , Nov 1970
Pages:773 - 776

[Abstract] [PDF Full-Text (632 KB)] **IEEE JNL**

5 **A technique to generate feasible tests for communications systems with multiple timers**
Fecko, M.A.; Uyar, M.U.; Duale, A.Y.; Amer, P.D.;
Networking, IEEE/ACM Transactions on , Volume: 11 , Issue: 5 , Oct. 2003
Pages:796 - 809

[Abstract] [PDF Full-Text (875 KB)] IEEE JNL

6 **A scenario-driven approach to trace dependency analysis**
Egyed, A.;
Software Engineering, IEEE Transactions on , Volume: 29 , Issue: 2 , Feb. 200
Pages:116 - 132

[Abstract] [PDF Full-Text (2830 KB)] IEEE JNL

7 **Exploring into programs for the recovery of data dependencies designed**
Hee Beng Kuan Tan; Tok Wang Ling; Cheng Hian Goh;
Knowledge and Data Engineering, IEEE Transactions on , Volume: 14 , Issue: 4 , July-Aug. 2002
Pages:825 - 835

[Abstract] [PDF Full-Text (355 KB)] IEEE JNL

8 **Modeling the impact of preflushing on CTE in proton irradiated CCD based detectors**
Philbrick, R.H.;
Nuclear Science, IEEE Transactions on , Volume: 49 , Issue: 2 , April 2002
Pages:559 - 567

[Abstract] [PDF Full-Text (383 KB)] IEEE JNL

9 **Automatic generation and targeting of application-specific operatin systems and embedded systems software**
Gauthier, L.; Sungjoo Yoo; Jerraya, A.A.;
Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on , Volume: 20 , Issue: 11 , Nov. 2001
Pages:1293 - 1301

[Abstract] [PDF Full-Text (136 KB)] IEEE JNL

10 **Using simulation for assessing the real impact of test-coverage on defect-coverage**
Briand, L.C.; Pfahl, D.;
Reliability, IEEE Transactions on , Volume: 49 , Issue: 1 , March 2000
Pages:60 - 70

[Abstract] [PDF Full-Text (272 KB)] IEEE JNL

11 **Harmonic spraying of conducting liquids employing AC-DC electric fields**
Huneiti, Z.A.; Balachandran, W.; Machowski, W.W.;

Industry Applications, IEEE Transactions on , Volume: 34 , Issue: 2 , March-Ap 1998
Pages:279 - 285

[Abstract] [PDF Full-Text (152 KB)] IEEE JNL

---

12 **The lazy sweep ray casting algorithm for rendering irregular grids**
Silva, C.T.; Mitchell, J.S.B.;
Visualization and Computer Graphics, IEEE Transactions on , Volume: 3 , Issue 2 , April-June 1997
Pages:142 - 157

[Abstract] [PDF Full-Text (1116 KB)] IEEE JNL

---

13 **Constraint meta-object: a new object model for distributed collaborative designing**
Ram, D.J.; Vivekananda, N.; Rao, C.S.; Mohan, N.K.;
Systems, Man and Cybernetics, Part A, IEEE Transactions on , Volume: 27 , Is 2 , March 1997
Pages:208 - 221

[Abstract] [PDF Full-Text (284 KB)] IEEE JNL

---

14 **Hierarchical execution to speed up pipeline interlock in mainframe computers**
Shintani, Y.; Shonai, T.; Kurokawa, H.; Kuriyama, K.; Yamaoka, A.;
Computers, IEEE Transactions on , Volume: 45 , Issue: 5 , May 1996
Pages:589 - 599

[Abstract] [PDF Full-Text (1080 KB)] IEEE JNL

---

15 **Functional requirements and cost-benefit study for distribution automation at B.C. Hydro**
Markushevich, N.S.; Herejk, I.C.; Nielsen, R.E.;
Power Systems, IEEE Transactions on , Volume: 9 , Issue: 2 , May 1994
Pages:772 - 781

[Abstract] [PDF Full-Text (672 KB)] IEEE JNL

---

1 2 3 4 5 6 7 Next

Terms used **automatically generate software code** Found **52** of **150,138**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

☐ Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 52 Result page: **1** 2 3 next

Relevance scale

1 Reuse in the application layer

Hirotomo Okuno, Hideki Matsumoto, Hironori Asai, Mikiko Sakurai, Takao Nakayama

November 1996 **Proceedings of the 1996 conference of the Centre for Advanced Studies on Collaborative research**

Full text available: pdf(151.98 KB) Additional Information: full citation, abstract, references, index terms

Today's advanced CASE tools, combining the building-block and generative approaches to software reuse, are effective for reuse of software components and procedures in the presentation layer and data layer of the three-layer model. There are no effective tools generally available, however, for reuse of application-layer components and procedures, which are usually too numerous and small for efficient reuse. Programmers therefore choose to transform existing specification sheets of the applicatio ...

2 Graphically-oriented reverse engineering tools for Ada software (abstract)

James H. Cross

January 1990 **Proceedings of the 1990 ACM annual conference on Cooperation**

Full text available: pdf(68.01 KB) Additional Information: full citation, abstract, index terms

Computer professionals have long promoted the idea that graphical representations of software are extremely useful as comprehension aids when used to supplement textual descriptions and specifications of software, especially for large complex systems. The general goal of this research is the study and formulation of graphical representations of algorithms, structures, and processes for Ada (GRASP/Ada). The research is presently focused on the extraction and generation of gr ...

3 System design methods: scheduling advances: Extended quasi-static scheduling for formal synthesis and code generation of embedded software

Feng-Shi Su, Pao-Ann Hsiung

May 2002 **Proceedings of the tenth international symposium on Hardware/software codesign**

Full text available: pdf(590.26 KB) Additional Information: full citation, abstract, references, citings, index terms

With the computerization of most daily-life amenities such as home appliances, the software in a real-time embedded system now accounts for as much as 70% of a system design. On one hand, this increase in software has made embedded systems more accessible and easy to use, while on the other hand, it has also necessitated further research on how complex embedded software can be designed automatically and correctly.

Enhancing recent advances in this research, we propose an *Extended Quasi-Static* ...

4 Time weaver: a software-through-models framework for embedded real-time systems
Dionisio de Niz, Raj Rajkumar
June 2003 **ACM SIGPLAN Notices , Proceedings of the 2003 ACM SIGPLAN conference on Language, compiler, and tool for embedded systems**, Volume 38 Issue 7
Full text available: pdf(467.76 KB) Additional Information: full citation, abstract, references, index terms

Embedded real-time systems are deployed in a wide range of application domains including transportation systems, automated manufacturing, process control, defense, aerospace, and telecommunications. These systems must satisfy not only logical functional requirements but also *para-functional* properties such as timeliness, Quality of Service (QoS) and reliability. The cross-cutting behaviors imposed by these para-functional properties and dependencies on operational characteristics (e.g. ha ...

Keywords: couplers, embedded, real-time, semantic dimension, semantic separation, software-through-models

5 Specification, verification, and synthesis of concurrency control components
Tuba Yavuz-Kahveci, Tevfik Bultan
July 2002 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 2002 ACM SIGSOFT international symposium on Software testing and analysis**, Volume 27 Issue 4
Full text available: pdf(315.84 KB) Additional Information: full citation, abstract, references

Run-time errors in concurrent programs are generally due to the wrong usage of synchronization primitives such as monitors. Conventional validation techniques such as testing become ineffective for concurrent programs since the state space increases exponentially with the number of concurrent processes. In this paper, we propose an approach in which 1) the concurrency control component of a concurrent program is formally specified, 2) it is verified automatically using model checking, and 3) the ...

Keywords: concurrent programming, infinite-state model checking, monitors, specification languages

6 Optimized unrolling of nested loops
Vivek Sarkar
May 2000 **Proceedings of the 14th international conference on Supercomputing**
Full text available: pdf(1.10 MB) Additional Information: full citation, abstract, references, citings, index terms

In this paper, we address the problems of automatically selecting unroll factors for perfectly nested loops, and generating compact code for the selected unroll factors. Compared to past work, the contributions of our work include a) a more detailed cost model that includes ILP and 1-cache considerations, b) a new code generation algorithm for unrolling nested loops that generates more compact code (with fewer remainder loops) than the unroll-and-jam transf ...

7 Generating parallel code from object oriented mathematical models
Niclas Andersson, Peter Fritzson
August 1995 **ACM SIGPLAN Notices , Proceedings of the fifth ACM SIGPLAN symposium on Principles and practice of parallel programming**, Volume 30 Issue 8
Full text available: pdf(1.01 MB) Additional Information: full citation, abstract, references, index terms

For a long time efficient use of parallel computers has been hindered by dependencies

introduced in software through low-level implementation practice. In this paper we present a programming environment and language called Object-Math (Object oriented Mathematical language for scientific computing), which aims at eliminating this problem by allowing the user to represent mathematical equation-based models directly in the system. The system performs analysis of mathematical models to extract ...

8 Automated and certified conformance to responsiveness policies

Joseph C. Vanderwaart, Karl Crary

January 2005 **Proceedings of the 2005 ACM SIGPLAN international workshop on Types in languages design and implementation**

Full text available: pdf(615.49 KB) Additional Information: full citation, abstract, references, index terms

Certified code systems protect computers from faulty or malicious code by requiring untrusted software to be accompanied by checkable evidence of its safety. This paper presents a certified code solution to a problem in grid computing, namely, controlling the CPU usage of untrusted programs. Specifically, we propose to endow the runtime system supervising local execution of grid programs with a trusted "yield" operation, and require the untrusted code to execute this operation with at least a ce ...

Keywords: certified code, grid computing, typed assembly language

9 Generating embedded software from hierarchical hybrid models

Rajeev Alur, Franjo Ivancic, Jesung Kim, Insup Lee, Oleg Sokolsky

June 2003 **ACM SIGPLAN Notices , Proceedings of the 2003 ACM SIGPLAN conference on Language, compiler, and tool for embedded systems**, Volume 38 Issue 7

Full text available: pdf(355.95 KB) Additional Information: full citation, abstract, references, index terms

Benefits of high-level modeling and analysis are significantly enhanced if code can be generated automatically from a model such that the correspondence between the model and the code is precisely understood. For embedded control software, *hybrid systems* is an appropriate modeling paradigm because it can be used to specify continuous dynamics as well as discrete switching between modes. Establishing a formal relationship between the mathematical semantics of a hybrid model and the actual ...

Keywords: code generation, embedded software, formal language, hybrid system, modularity

10 Derive: a tool that automatically reverse-engineers instruction encodings

Dawson R. Engler, Wilson C. Hsieh

January 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN workshop on Dynamic and adaptive compilation and optimization**, Volume 35 Issue 7

Full text available: pdf(1.07 MB) Additional Information: full citation, abstract, references, citings, index terms

Many binary tools, such as disassemblers, dynamic code generation systems, and executable code rewriters, need to understand how machine instructions are encoded. Unfortunately, specifying such encodings is tedious and error-prone. Users must typically specify thousands of details of instruction layout, such as opcode and field locations values, legal operands, and jump offset encodings. We have built a tool called DERIVE that extracts these details from existing software: the system assemble ...

11 Model checking without a model: an analysis of the heart-beat monitor of a telephone switch using VeriSoft

Patrice Godefroid, Robert S. Hanmer, Lalita Jategaonkar Jagadeesan

March 1998 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 1998 ACM**

SIGSOFT international symposium on Software testing and analysis, Volume 23 Issue 2

Full text available: pdf(1.15 MB) Additional Information: full citation, abstract, references, citings, index terms

VeriSoft is a tool for systematically exploring the state spaces of systems composed of several concurrent processes executing arbitrary code written in full-fledged programming languages such as C or C++. The state space of a concurrent system is a directed graph that represents the combined behavior of all concurrent components in the system. By exploring its state space, VeriSoft can automatically detect coordination problems between the processes of a concurrent system.We report in this pape ...

12 Automated application programming environment

Yap S. Chua, Charles N. Winton

May 1986 **ACM SIGAPL APL Quote Quad , Proceedings of the international conference on APL**, Volume 16 Issue 4

Full text available: pdf(964.57 KB) Additional Information: full citation, abstract, references, index terms

With the availability of high-performance, low-cost hardware, APL provides a cost-effective means of developing custom software for the small business environment, and perhaps the best alternative to trying to adapt to an off-the-shelf package. Since APL does not inherently deal with system specific screen manipulation and file management techniques, programming these interface requirements from APL is typically tedious, potentially reducing the attractiveness of APL as the development lang ...

13 Demos: QuickUML: a tool to support iterative design and code development

Carl Alphonce, Phil Ventura

October 2003 **Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**

Full text available: pdf(76.31 KB) Additional Information: full citation, abstract, index terms

We demonstrate QuickUML, a tool which supports iterative design and code development by providing facilities to draw UML class diagrams, to generate Java code from such diagrams, and also to automatically generate a UML class diagram from a collection of Java source code files. We also discuss how use of the tool provides general support for teaching students the importance of design in software development.

Keywords: CS1, CS2, Java, UML, design, object-orientation, tool

14 Operating Systems: Remote customization of systems code for embedded devices

Sapan Bhatia, Charles Consel, Calton Pu

September 2004 **Proceedings of the fourth ACM international conference on Embedded software**

Full text available: pdf(208.16 KB) Additional Information: full citation, abstract, references, index terms

Dedicated operating systems for embedded systems are fast being phased out due to their use of manual optimization, which provides high performance and small footprint, but also requires high maintenance and portability costs every time hardware evolves.In this paper, we describe an approach based on customization of generic operating system modules. Our approach uses a remote customization server to automatically generate highly optimized code that is then loaded and executed in the kernel of t ...

Keywords: remote customization, remote specialization, specialization server

15 Automatic Generation of a Real-Time Operating System for Embedded Systems

Felice Balarin, Massimiliano Chiodo, Attila Jurecska, Luciano Lavagno, Bassam Tabbara, Alberto Sangiovanni-Vincentelli
March 1997 **Proceedings of the 5th International Workshop on Hardware/Software Co-Design**
Full text available: Publisher Site Additional Information: full citation, abstract

Embedded systems are typically implemented as a set of communicating components some of which are implemented in hardware and some of which are implemented in software. Usually many software components share a processor. A real-time operating system (RTOS) is used to enable sharing and provide a communication mechanism between components. Commercial RTOSs are available for many popular micro-controllers. Using them provides significant reduction in design time and often leads to better structure ...

Keywords: co-synthesis, Real-Time Operating Systems, scheduling

16 Compiler analysis and optimization: Static program analysis of embedded executable assembly code
Ramakrishnan Venkitaraman, Gopal Gupta
September 2004 **Proceedings of the 2004 international conference on Compilers, architecture, and synthesis for embedded systems**
Full text available: pdf(213.93 KB) Additional Information: full citation, abstract, references, index terms

We consider the problem of automatically checking if coding standards have been followed in the development of embedded applications. The problem arises from practical considerations because DSP chip manufacturers (in our case Texas Instruments) want various third party software developers to adhere to a certain coding standard to facilitate system integration during application development. Checking for compliance with coding standards, in general, is undecidable. Moreover, only machine code of ...

Keywords: abstract interpretation, assembly code, embedded software components, executable code, static analysis

17 Embedded software: Embedded software generation from system level design languages
Haobo Yu, Rainer Dömer, Daniel Gajski
January 2004 **Proceedings of the 2004 conference on Asia South Pacific design automation: electronic design and solution fair 2004**
Full text available: pdf(108.36 KB) Additional Information: full citation, abstract, references

To meet the challenge of increasing design complexity, designers are turning to system level design languages (SLDLs) to model systems at a higher level of abstraction. This paper presents a method of automatically generating embedded software from system specification written in SLDL. Several refinement steps and intermediate models are introduced in our software generation flow. We demonstrate the effectiveness of the proposed method by a tool which can generate efficient ANSI C code from syst ...

18 Session 6B: Software reuse: ODGen: a prototype reverse engineering tool for Ada
Kelly Morrison
April 1992 **Proceedings of the 30th annual Southeast regional conference**
Full text available: pdf(405.47 KB) Additional Information: full citation, abstract, references

The problem of keeping documentation current with an evolving software system has always been a major concern in software engineering, especially as deriving an understanding of a program's logic may represent as much as 90% of the cost of maintenance. Reverse engineering, the processing of source code to extract higher levels of

abstraction, promises to be a viable solution to this problem as tools may be developed to automatically extract design information from a software system. This paper d ...

19 Generating Tests from Counterexamples

May 2004 **Proceedings of the 26th International Conference on Software Engineering**

Full text available: pdf(263.07 KB) Publisher Site Additional Information: full citation, abstract

We have extended the software model checker BLAST toautomatically generate test suites that guarantee full coveragewith respect to a given predicate. More precisely, givena C program and a target predicate p, BLAST determinesthe set L of program locations which program execution canreach with p true, and automatically generates a set of testvectors that exhibit the truth of p at all locations in L. Wehave used BLAST to generate test suites and to detect deadcode in C programs with up to 30 K lin ...

20 Potpourri: HOIST: a system for automatically deriving static analyzers for embedded systems

John Regehr, Alastair Reid

October 2004 **Proceedings of the 11th international conference on Architectural support for programming languages and operating systems**

Full text available: pdf(145.53 KB) Additional Information: full citation, abstract, references, index terms

Embedded software must meet conflicting requirements such as be-ing highly reliable, running on resource-constrained platforms, and being developed rapidly. Static program analysis can help meet all of these goals. People developing analyzers for embedded object code face a difficult problem: writing an abstract version of each instruction in the target architecture(s). This is currently done by hand, resulting in abstract operations that are both buggy and im-precise. We have developed Hoist: a ...

Keywords: abstract interpretation, object code, program verification, static analysis

Results 1 - 20 of 52 Result page: **1** 2 3 next